

PATENT

Attorney Docket No. ADHE1100

APPLICATION FOR U.S. PATENT
TRANSMITTAL FORM

THE COMMISSIONER OF PATENTS
AND TRADEMARKS
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of:

Inventor(s): Aubrey McAuley

Filing Date: October 1, 1999

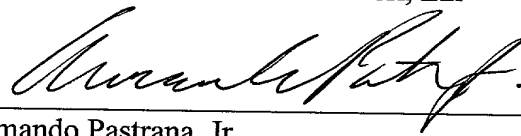
For: SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY
OBJECT FRAMEWORK

Enclosed are: 2 Sheets of Formal Drawings

FEE CALCULATION					FEE
	Number of Claims	Number of Allowed Claims	Number Extra	RATE	BASIC FEE \$ 380.00
Total Claims	53	-20	33	X \$9 =	\$ 297.00
Independent Claims		- 3	0	X \$39 =	\$ 0.00
TOTAL FILING FEE =					\$677.00

The Asst. Commissioner of Patents is hereby authorized to deduct **\$677.00**
representing the appropriate filing fee for the above-identified patent application, from
Deposit Account No. 50-0456 of Gray Cary Ware & Freidenrich, LLP. *A duplicate copy of
the paper is enclosed.*

GRAY CARY WARE ▲ FREIDENRICH, LLP



Armando Pastrana, Jr.
Attorney for Applicant

Registration Number: P-44,997

Dated: October 1, 1999

Attorney Docket No.: ADHE1100
Applicant or Patentee: Aubrey McAuley
Filed: Herewith
Title: SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS
(37 C.F.R. §§ 1.9(f) and 1.27(c)) - SMALL BUSINESS CONCERN

I hereby declare that I am

- ☐ the owner of the small business concern identified below:
☒ an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF CONCERN: Adhesive Software
ADDRESS OF CONCERN: 800 Brazos, Suite 1040
Austin, Texas 78701

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 C.F.R. §121.3-18 and reproduced in 37 C.F.R. § 1.9(d), for purposes of paying reduced fees under Section 41(a) and (b) of Title 35, United States Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention, entitled System and Method for Generating Web Sites in an Arbitrary Object Framework by inventor Aubrey McAuley, described in:

- ☒ the specification filed herewith with title as listed above.
☐ the application identified above.
☐ the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights to the invention is listed below and no rights to the invention are held by any person, other than the inventor, who could not qualify as a small business concern under 37 C.F.R. §1.9(d) or by any concern which would not qualify as a small business concern under 37 C.F.R. §1.9(d) or a nonprofit organization under 37 C.F.R. §1.9(e).

NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities (37 C.F.R. §1.27).

Full Name _____
Address _____
☐ Individual ☒ Small Business Concern ☐ Nonprofit Organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate (37 C.F.R. §1.28(b)).

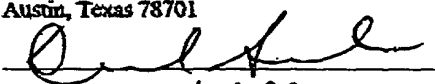
I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING: Derek Saunders
TITLE OF PERSON OTHER THAN OWNER: General Counsel
ADDRESS OF PERSON SIGNING: 800 Brazos, Suite 1040
Austin, Texas 78701

SIGNATURE: _____

DATE: _____

ALW023236.1
104304-991100


10-1-99

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In the Application of: Aubrey McAuley
Date Filed: October 1, 1999
Title: SYSTEM AND METHOD FOR GENERATING WEB SITES IN AN
ARBITRARY OBJECT FRAMEWORK


ATTN: BOX PATENT APPLICATION

Hon. Assistant Commissioner of Patents
Washington, D.C. 20231

Dear Sir:

CERTIFICATE OF MAILING BY EXPRESS MAIL

I hereby certify that the attached Transmittal Form, Verified Statement Claiming Small Entity Status, Patent Application, Declaration and Power of Attorney, Drawings, and postcard are being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. 1.10 on this 31st day of October, 1999, and is addressed to the Assistant Commissioner of Patents, Washington, D.C. 20231 Attn: Box Patent Applications.


Janice Di Petta

Express Mail Receipt No.:
EL409604058US
Attorney's Docket:
ADHE1100

SYSTEM AND METHOD FOR GENERATING
WEB SITES IN AN ARBITRARY OBJECT FRAMEWORK

TECHNICAL FIELD OF THE INVENTION

5 This invention relates generally to systems and
methods for generating software applications in an
arbitrary object framework, and more specifically to
systems and methods for generating web sites in an
arbitrary object framework.

10

65T00T"4EE0T460

BACKGROUND OF THE INVENTION

Three processes used to create complex software applications such as web sites are form, function, and content. Form includes graphic designs, user
5 interfaces, and graphical representations created by a designer or a group of designers. Function includes logical functionality, which can be software code created by a programmer or group of programmers. Form includes informative content. Informative content can
10 include written, recorded, or illustrated documentation, such as photographs, illustrations, product marketing material, and news articles. Content can be created by writers, photographers, artists, reporters, or editors.

15 Currently, typical workflows dictate a serial approach to integrating the form, function, and content to create complex software applications such as a web site. The serial approach is illustrated in FIGURE 1. In FIGURE 1, content 10 for a complex software
20 application can be chosen or created. Form 12 for the presentation of content 10 can then be created. Functionality 14 can then be generated using code to create the complex software application (product 16) with the desired information (content 10) and style
25 (form 12). Using the method illustrated in FIGURE 1, every final component of the complex software application must be manipulated by a programmer before it is ready to be used. The exact workflow may vary from industry to industry or business to business, but
30 the basic restrictions are generally the same.

A traditional approach such as that illustrated in
FIGURE 1, may create unwanted bottlenecks in the
production process. Each upstream revision, such as a
change of content 10 or design 12, forces a repetition
of the entire process. As an example, consider a web
site for a large newspaper. The web site may have a
function that can include a file into the web site.
The marketing department may decide to change the
appearance of the header on the web site depending on
the browser of a user. In this case, a programmer may
need to invoke an external script or embed some
specific logic within the web site. Unfortunately, if
there is a large web site with thousands of pages of
information stored on a server, the programmer may have
to change every one of the thousands of pages.
Therefore, a small change by the marketing department
can cause a large burden on the programming department.

Prior art solutions have succeeded in partially
separating some of these functions. Notably, content
management databases and digital repositories provide a
means of separating content from form and function.
Likewise, sophisticated software development teams
frequently employ internal code structuring techniques
that can help to minimize dependencies between
interface designs and the functions they access.
However, content management tools typically fail to
address form/function issues. Therefore, there can
still be production slow-downs due to changes in form
that require a subsequent change in functionality.

SUMMARY OF THE INVENTION

Therefore a need exists for a method of generating complex software applications that reduces or eliminates production delays and the workload for programmers due to changes in content and/or form. This method should separate form, content and function so that each area can be independently changed.

The present invention provides a system and method for generating software applications that substantially eliminates or reduces disadvantages and problems associated with previously developed systems and methods used for generation of software applications. More specifically, the present invention provides a method for generating software applications in an arbitrary object framework. The method of the present invention separates content, form, and function of the computer application so that each may be accessed or modified independently. The method of this invention includes creating arbitrary objects, managing the arbitrary objects throughout their life cycle, and deploying the arbitrary objects in a design framework for use in complex computer applications.

The present invention provides an important technical advantage in that content, form, and function are separated from each other in the generation of the software application. Therefore, changes in design or content do not require the intervention of a programmer. This advantage decreases the time needed to change various aspects of the software application.

Consequently, cost is reduced and versatility is increased.

5 The present invention provides another technical advantage in that users are not required to use a proprietary language to encode. These arbitrary objects may include encapsulated legacy data, legacy systems and custom programming logic from essentially any source in which they may reside. Any language supported by the host system, or any language that can
10 be interfaced to by the host system, can be used to generate an object within the application.

The present invention provides yet another technical advantage in that it can provide a single point of administrative authority that can reduce
15 security risks. For instance, a large team of programmers can work on developing a large group of arbitrary objects within the object library. If one object has a security hole, an administrator can enter the object library and disable that arbitrary object.

20 Still another technical advantage of the present invention is that it enables syndication of the software application. As noted above, functionality is separate from form and content. Consequently, a user can easily introduce a new look for the application or
25 syndicate the content and functionality of the application to another group without having to recode all of the objects needed to access content.

Another technical advantage of the present invention is that it allows for personalization and
30 profiling. With personalization, the web presentation

is tailored to the specific needs of the web user based
on the user's past history. Profiling also enables
tailoring a web site or presentation. Profiling is
dependent on environmental variables such as browser
type or IP address.

5

667007 4E0E460

BRIEF DESCRIPTION OF THE DRAWINGS

A more complete understanding of the present invention and the advantages thereof may be acquired by referring to the following description, taken in
5 conjunction with the accompanying drawings in which like reference numbers indicate like features and wherein:

FIGURE 1 illustrates a prior art workflow diagram for generating a software product;

10 FIGURE 2 is a hierarchical workflow diagram for one embodiment of the present invention;

FIGURE 3 is a flow diagram for one embodiment of the present invention;

15 FIGURE 4 is a flow diagram for the embodiment illustrated in FIGURE 4.

FIGURE 5 is a diagram illustrating the components of one embodiment of the present invention used to generate web sites; and

DETAILED DESCRIPTION OF THE INVENTION

Preferred embodiments of the present invention are illustrated in the FIGURES, like numerals being used to refer to like and corresponding parts of various drawings.

The present invention provides a system and method for using a hierarchical, arbitrary object framework for generating software applications. The method separates content, form, and function of the software application so that each can be accessed or modified independently. The method of this invention includes creating arbitrary objects, managing the arbitrary objects in an object library, and deploying the arbitrary objects in a design framework for use in computer applications.

FIGURE 2 is a hierarchical workflow diagram for the present invention. Product 6 includes three contributing groups: content 10, form 12, and functionality 14. Content 10 can include written, recorded, or illustrated collateral such as documentation, photographic illustrations, product marketing material, and articles. Form 12 can include graphic designs such as user interfaces and graphical presentations. Function 14 can include the logical functionality of software code and scripts. The hierarchical framework separates content 10, form 12, and functionality 14 to generate product 16. Product 16 may be a computer software application such as a web site. Since content 10, design 12, and functionality 14 are separate entities independent of each other,

modification in one group does not require corresponding modifications in another group. Each group can contribute to product 16 directly.

FIGURE 3 is a flow diagram of one embodiment of the present invention. At step 20, arbitrary objects can be generated. Arbitrary objects may include any combination of application logic and data desired by a developer. Arbitrary objects can include text file pointers, binary file pointers, compiled executables, scripts, data base queries, shell commands, remote procedure calls, global variables, and local variables. The arbitrary object framework allows arbitrary objects to be referenced in a consistent manner regardless of the type. Also, the arbitrary object framework allows local arbitrary objects to either override global parent arbitrary objects or inherit capabilities and data from the global parent, regardless of the type of the local arbitrary object.

At step 22, these arbitrary objects can be managed in an object library. The life cycle of these objects may be managed in a consistent manner using revision tracking, roll back, and sign off. At step 24, objects can be deployed from the object library into a design framework to create the software application. Because the object pointers are not tied in any way to the functionality of the object, an object of one type can be easily replaced with another object of another type. This eliminates a common problem in content management systems of the inability to preview content within its appropriate location on the site or within the system.

Normally, a special system made for the purpose of previewing a piece of content would have to be hard-coded to view the current approved live content for all other pieces except the piece in question. This multiplies the design problem, because changes in the design in the main site change all previous templates. In the method of the present invention, since all that exists within the framework is an arbitrary object, the arbitrary object can be swapped for another object that pulls the current piece content in question.

Using one embodiment of this invention, for example, the Features or Editorials page of a newspaper can be dynamically replaced. The present invention can execute all the normal objects that can be placed on the page to show the content as it would appear, and then take the one piece in question and replace it with a second object to be examined. Objects may be deployed globally across an entire system or locally within a specific area or sub-areas of a system.

FIGURE 4 represents a flow diagram of another embodiment of the present invention. At step 30, arbitrary objects can be generated. At step 32, the arbitrary objects can be managed in an object library. Arbitrary objects can be deployed in a container page at step 34 to generate a web site.

Arbitrary objects may include any combination of application logic and data desired by a developer. Arbitrary objects can include text file pointers, binary file pointers, compiled executable scripts, database queries, shell commands, remote call

procedures, global variables and local variables. Arbitrary objects may also include cached data queries and executables. The arbitrary object framework allows arbitrary objects to be referenced in a consistent manner regardless of the type of object. Also, the arbitrary object framework allows local arbitrary objects to either override global parent arbitrary objects or inherit capabilities and data from the global parent arbitrary object.

Arbitrary objects can execute any function that can be run or understood by the host computer system so that any underlying functionality of the operating system used by the host system can be defined as an object within the arbitrary framework. Legacy data, document objects, CPI programs, and database queries can all be encapsulated as objects within the arbitrary framework. The arbitrary object can be accessed by an arbitrary object name. Arbitrary objects are not tied to their functionality. One arbitrary object can be easily replaced with another arbitrary object of another type.

Arbitrary objects can be managed in an object library. The life cycle of the arbitrary objects may be managed in a consistent manner using revision tracking, roll-back, and sign-off. The object library can include separate specialized object libraries that can be administered separately by different developers in each area. For instance, for a web site used to generate a newspaper, there may be an advertising object library that is physically distinguished from

other object libraries, such as an object library for sports or an object library for news. Therefore, queries for advertising can be created without impacting any other area of the web site.

5 Arbitrary objects can be deployed from the object library into a container page to generate the web site. The container page is a truly dynamic page. Unlike prior art methods, where a static copy of information is often pushed over a firewall to a live web site, the
10 present invention incorporates object caching. An arbitrary object can be cached, rather than caching an entire page. When the arbitrary object is cached, certain elements of the arbitrary object can be specified as dynamic elements while others can be
15 specified as static elements. Therefore, a web site can contain multiple dynamic web pages wherein objects used to construct the form, function, and content of the web page can contain dynamic elements and static elements. This provides flexibility for what needs to
20 be computed or processed at the time that someone, such as a web user, accesses the web page.

FIGURE 5 shows the components of one embodiment of the present invention used to generate web sites. A user with web browser 40 can connect to web server 44
25 through internet or intranet 42. Web server 44 can access static HTML web documents 46 as well as dynamic HTML documents 52. Dynamic HTML web documents 52 can be created using WebOS Object Manager 50. Dynamic HTML Web document 52 can include document objects 56, shell
30 scripts 58, CGI programs 60, and database queries 62.

Document objects 56, shell scripts 58, CGI programs 60, and database queries 62 can be stored in WebOS object library 54. Database queries 62 can result from extracting information from WebOS Information Database 68 and inputting the information into Dynamic HTML Web Template 66.

User Profile and Password Database 70 can provide web sites or systems with a means to take advantage of customer profiles to look at customer preferences or history, and dynamically replace a website object with another object that contains content information matching the user profile or preferences. Thus, the web site or system can dynamically allocate the correct content for a customer. This is important in commerce applications. A customer's buying history can be examined for trend items and the customer presented products that match his or her profile. Present personalization systems are written purely in custom code and require an inordinately large amount of time to construct the custom applications necessary to interpret the preferences of an individual user.

The method of present invention can perform object caching. This means that an object can be cached instead of caching an entire page. Object caching permits specifying elements of an object to be dynamic and elements of the object to be static. A system user can thus have the flexibility of specifying what needs to be computed or processed at the time a user accesses the system versus trying to anticipate and calculate in advance and cache and post the object over to a server.

Many functions are stored within an object library on an arbitrary object framework such that those functions can be accessed by name arbitrarily. This is in contrast to a traditional model where the function must be explicitly invoked with all its parameters included. Objects may execute any function that can be run or understood by the host computer system so that any underlying functionality of the host's operating system can be defined as an object within the framework of the method of the present invention. The object library can contain legacy data, document objects, CTI programs, and/or database queries, that can all be encapsulated as objects within a framework and accessed from within a design. All that is needed is the name of the function in order to access the function.

Objects can be controlled to perform functions based on a profile of an individual and environmental variables, such as the type of browser, the country of the individual or the individual's IP address. A specific competitor may be blocked from seeing certain objects on a web page created using the method of the present invention.

A critical distinction between the present invention and previous object oriented development systems is the need to know how a function can be called and what to expect it to return, rather than just knowing the function's name. This means that typically the system administrator calls the name of an object and passes parameters to the object. Any and all variable information or environmental information

can be available to every object. The environment space can be available to all objects executed and an object can arbitrarily take advantage of any of the environmental information, depending on the design of the object.

Different areas of a web site can be administered separately by different developers in each of these areas. An advertising object library can be physically distinguished from other object libraries, such as those for sports and news. An advertising programmer can create new queries for the advertising section of a site without having to worry about affecting other areas of the site.

The present invention allows different object types to be interchangeable. The object name is essentially just another variable in the environment. Also different variables can also be interchangeable. The object framework can be designed such that objects and variables can be kept in the same name space, every object can have access to all the environmental settings, and every object pointer can potentially be another name in the name space.

Object caching, rather than page caching can be implemented with the present invention. These objects can be stored in an object library. An object in the object library can be a file, a global variable, an executable script, a database query, a cached executable or a cached database query. This means that the results of a query can be stored in a static file using the object name as long as the static file has

not expired. This is important if the query is a lengthy query.

A technical advantage of the present invention is that it allows for syndication. Syndication enables
5 the content and function of a particular web site to be syndicated to another web site or web presentation. For instance, if a company would like to roll out a new look or syndicate its content and functionality to another business, this can be easily accomplished using
10 the present invention. Since there is no application code resident in a web page itself, the same data can be repackaged in a number of different ways across multiple sites. There is no need to recode the design elements or design pages on the web site or recode any
15 functions that are needed to access the content of the website. The present invention enables electronic store fronts to sell from a single source with a unique interface design. Also, newspaper chains can distribute international and national content from a
20 single source and add local content themselves.

Another technical advantage of the present invention is that it allows for a single point of control when developing a web site. Therefore, if a large team of developers are working on a site, and
25 multiple persons are contributing arbitrary objects to the overall arbitrary framework, then if one of the arbitrary objects has a security hole in it, the arbitrary object can be easily accessed in the object library and disabled. This security feature can

immediately shut down that function across the entire web site and patch the security hole.

The present invention provides still another technical advantage in that it allows for personalization. Personalization enables companies that want to take advantage of a customer profile to look at the customer's preferences or histories and deploy information to the web site specific to the customer.

Another technical advantage of the present invention allows for profiling. Profiling enables control over the arbitrary objects presented in a web site based on a profile of the individual accessing the web site. Profiling entails determining different environmental variables such as the type of browser hitting the site, the country of the individual accessing the site, and/or the individual's IP address. This can enable a company to present specific information to the individual based on the individual's environmental variables.

Although the present invention has been described in detail herein with reference to the illustrative embodiments, it should be understood that the description is by way of example only and is not to be construed in a limiting sense. It is to be further understood, therefore, that numerous changes in the details of the embodiments of this invention and additional embodiments of this invention will be apparent to, and may be made by, persons of ordinary skill in the art having reference to this description.

It is contemplated that all such changes and additional embodiments are within the spirit and true scope of this invention as claimed below.

551001-EE01460

WHAT IS CLAIMED IS:

1. A method for generating computer application
on a host system in an arbitrary object framework that
separates a content of said computer application, a
5 form of said computer application and a functionality
of said computer applications, said method comprising:

creating arbitrary objects with corresponding
arbitrary names of various object types for generating
said content of said computer application, said form of
10 said computer application, and said functionality of
said computer application;

managing said arbitrary objects in an object
library; and

deploying said arbitrary objects from said object
15 library into a design framework to create said software
application.

2. The method of Claim 1, wherein said computer
application is a web site.

3. The method of Claim 1, wherein said various
object types comprise text file pointers.

4. The method of Claim 1, wherein said various
25 object types comprise binary file pointers.

5. The method of Claim 1, wherein said various
object types comprise compiled executables.

6. The method of Claim 1, wherein said various object types comprise shell commands.

5 7. The method of Claim 1, wherein said various object types comprise remote procedure calls.

8. The method of Claim 1, wherein said various object types comprise global variables.

10 9. The method of Claim 1, wherein said various object types comprise cached executables.

15 10. The method of Claim 1, wherein said various object types comprise cached database queries.

11. The method of Claim 1, wherein said various object types comprise local variables.

20 12. The method of Claim 1, wherein said various object types comprise local objects and global parent objects.

25 13. The method of Claim 12, wherein said local objects can override said global parent objects.

14. The method of Claim 12, wherein said local objects inherit data from said global parent objects.

21

15. The method of Claim 12, wherein said local objects inherit capabilities from said global parent objects.

5 16. The method of Claim 1, further comprising deploying arbitrary objects globally.

17. The method of Claim 1, further comprising deploying arbitrary objects locally.

10 18. The method of Claim 1, wherein the step of managing said arbitrary objects further comprises using revision tracking.

15 19. The method of Claim 1, wherein the step of managing said arbitrary objects further comprises using rollback.

20 20. The method of Claim 1, wherein the step managing further comprises using signoff.

25 21. The method of Claim 1, wherein said arbitrary objects can be accessed and deployed into said design framework using said corresponding arbitrary names.

22. The method of Claim 1, further comprising swapping an arbitrary object of one type with an arbitrary object of another type.

23. The method of Claim 1, further comprising
caching objects.

5 24. The method of Claim 23, wherein the step of
caching objects further comprises specifying some
elements of an arbitrary object to be dynamic elements
and specifying some elements of said arbitrary object
to be static elements.

10 25. The method of Claim 1, further comprising
generating arbitrary objects in a programming language
that is compatible or supported by said host system.

651001 "EEOT" 60

26. A method for generating a web site on a host
system in an arbitrary object framework that separates
a content of said web site, a form of said web site,
and a functionality of said web site, said method
5 comprising:

creating arbitrary objects with corresponding
arbitrary names of various object types for generating
said content of said web site, said form of said web
site, and said functionality of said web site;

10 managing said arbitrary objects in an object
library; and

deploying said arbitrary objects from said object
library to a container page to create said web site.

15 27. The method of Claim 26, wherein said various
object types comprise text file pointers.

28. The method of Claim 26, wherein said various
object types comprise binary file pointers.

20 29. The method of Claim 26, wherein said various
object types comprise compiled executables.

25 30. The method of Claim 26, wherein said various
object types comprise shell commands.

31. The method of Claim 26, wherein said various
object types comprise remote procedure calls.

32. The method of Claim 26, wherein said various object types comprise global variables.

5 33. The method of Claim 26, wherein said various object types comprise local variables.

10 34. The method of Claim 26, wherein said various object types comprise local objects and global parent objects.

15 35. The method of Claim 34, wherein said local objects can override said global parent objects.

36. The method of Claim 34, wherein said local objects inherit data from said global parent objects.

20 37. The method of Claim 34, wherein said local objects inherit capabilities from said global parent objects.

38. The method of Claim 26, further comprising deploying arbitrary objects globally.

25 39. The method of Claim 26, further comprising deploying arbitrary objects locally.

30 40. The method of Claim 26, wherein the step of managing said arbitrary objects further comprises using revision tracking.

41. The method of Claim 26, wherein the step of managing said arbitrary objects further comprises using rollback.

5 42. The method of Claim 26, wherein the step managing said arbitrary objects further comprises using signoff.

10 43. The method of Claim 26, wherein said arbitrary objects can be accessed and deployed into said container page using said corresponding arbitrary names.

15 44. The method of Claim 26, further comprising swapping an arbitrary object of one type with an arbitrary object of another type.

20 45. The method of Claim 26, further comprising caching objects.

25 46. The method of Claim 45, wherein the step of caching objects further comprises specifying some elements of an arbitrary object to be dynamic elements and specifying some elements of said arbitrary object to be static elements.

30 47. The method of Claim 26, further comprising generating arbitrary objects in a programming language that is compatible or supported by said host system.

48. The method of Claim 26, wherein said various object types comprise cached executables.

5 49. The method of Claim 26, wherein said various object types comprise cached database queries.

50. The method of Claim 26, further comprising profiling of a user accessing said web site.

10 51. The method of Claim 26, further comprising personalization of said web site for a user accessing said web site.

15 52. The method of Claim 26, wherein said container page comprises arbitrary objects with both dynamic and static elements.

20 53. The method of Claim 26, wherein said content of said web site and said function of said web site can be syndicated.

554001-1000000

SYSTEM AND METHOD FOR GENERATING
SOFTWARE APPLICATION IN AN ARBITRARY OBJECT FRAMEWORK

ABSTRACT OF THE INVENTION.

5 A system and method for generating computer
applications in an arbitrary object framework. The
method separates content, form, and function of the
computer application so that each may be accessed or
modified separately. The method includes creating
10 arbitrary objects, managing the arbitrary objects
throughout their life cycle in an object library, and
deploying the arbitrary objects in a design framework
for use in complex computer applications.

661001-1100

FIG. 1
(PRIOR ART)

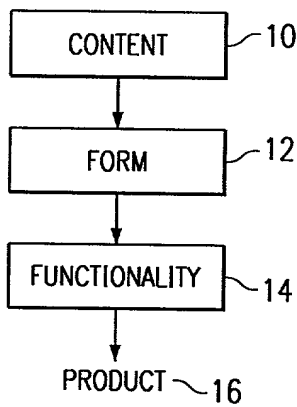


FIG. 2

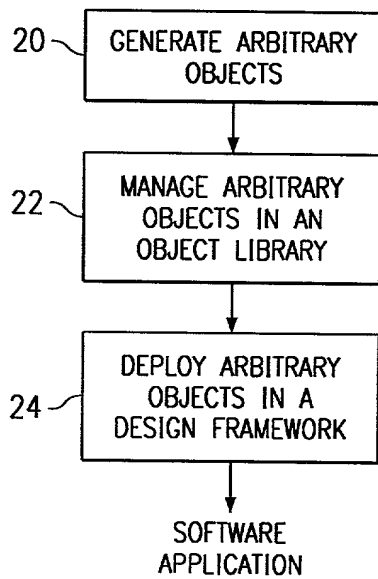
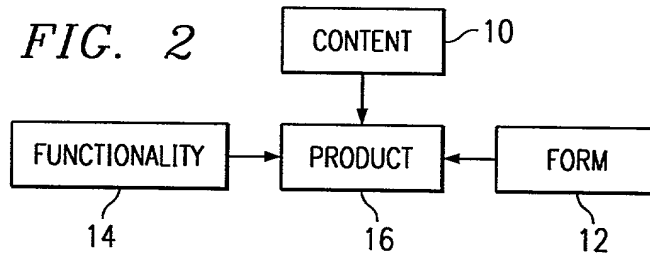


FIG. 3

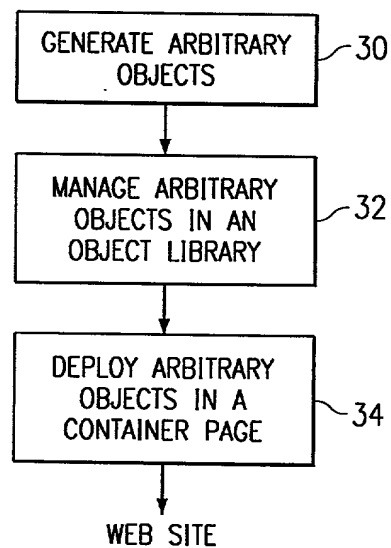
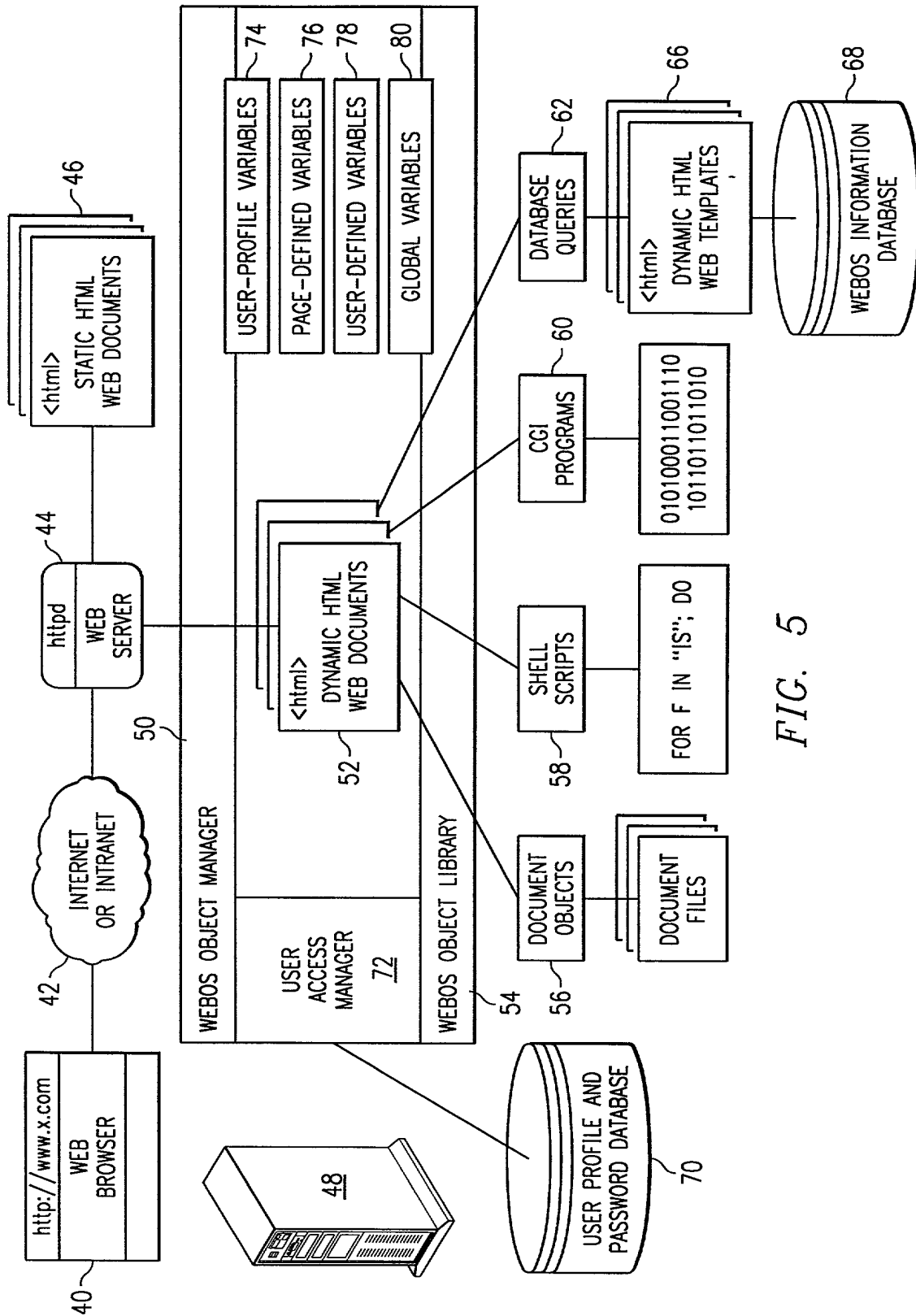


FIG. 4



DECLARATION AND POWER OF ATTORNEY

As the below-named inventor(s), I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if plural, names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention, design or discovery entitled **System and Method for Generating Web Sites in an Arbitrary Object Framework**, the specification of which (check one):

 X is attached hereto.
 was filed on _____ as Application
Serial No. _____ and was amended on
_____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment(s) referred to above; that I do not know and do not believe that said invention, design or discovery was ever known or used in the United States of America before my invention or discovery thereof, or patented or described in any printed publication in any country before my invention or discovery thereof, or more than one year prior to this application, or in public use or on sale in the United States of America more than one year prior to this application; that said invention, design or discovery has not been patented or made the subject of an inventor's

certificate issued prior to the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns; and that I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me which is material to the patentability as defined in 37 C.F.R. § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

<u>Number</u>	<u>Country</u>	<u>Date Filed</u>	<u>Priority Claimed</u>
			(Yes) (No)

I hereby claim the benefit under Title 35, United States Code, § 119(e) of any United States provisional application(s) listed below.

(Serial No.)

(Filing Date)

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application(s) in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to me to be material to

patentability as defined in 37 C.F.R. § 1.56 which became available between the filing date of the prior application(s) and the national or PCT international filing date of this application:

<u>Application</u> <u>Serial Number</u>	<u>Date Filed</u>	<u>Status</u>
--	-------------------	---------------

None

I hereby appoint:

WILLIAM N. HULSEY III	Registration No. 33,402
STEVEN R. SPRINKLE	Registration No. 40,825
STEPHEN E. REITER	Registration No. 31,192
TERRANCE A. MEADOR	Registration No. 30,298
GREGORY P. RAYMER	Registration No. 36,647
DAVID F. KLEINSMITH	Registration No. 40,050
BARRY N. YOUNG	Registration No. 27,774
TIMOTHY W. LOHSE	Registration No. 35,255
STANLEY H. KIM	Registration No. 40,047
DARLENE W. HAYES	Registration No. 33,899
RAMSEY R. STEWART	Registration No. 38,322
ROBERT A. MCCLAUCHLAN	Registration No. P,44,924
ARMANDO PASTRANA, JR.	Registration No. P,44,997

all of the firm of Gray Cary Ware & Freidenrich, as my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the United States Patent and Trademark Office connected therewith, and to file and prosecute any international patent applications filed thereon before any international authorities and to file any continuation, divisional, continuation-in-part, reissue or re-examination applications thereon.

Direct all telephone calls to:

Armando Pastrana, Jr.
Telephone: (512) 457-7080

Send all correspondence to:

Armando Pastrana, Jr.
GRAY CARY WARE ▲ FREIDENRICH LLP
100 Congress Avenue
Suite 1440
Austin, TX 78701-4042

I hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of first inventor:	Aubrey nmi McAuley
Inventor's signature:	_____
Date:	_____
Residence (City, County, State)	Austin, Travis, Texas
Citizenship:	U.S.A.
Post Office Address:	3810 Stonecroft Drive Austin, Texas 78749